15. VA FileMan Functions (Creating)

INTRODUCTION

As mentioned in the "VA FileMan Functions" chapter of the *VA FileMan Advanced User Manual*, as a programmer in FileMan you can create your own computed-expression functions. In some ways, a function can be thought of as an OUTPUT transform that can work on any field. For example, you may have a preference for seeing many dates displayed as 20-7-69, rather than the JUL 20,1969 format that FileMan typically produces. Since this date is internally stored in the form 2690720 (see the description of %DT), you could write a line of code that took the internally stored format in the variable X and transformed it using:

```
+$E(X,6,7)_"-"_+$E(X,4,5)_"-"_$E(X,2,3)
```

FUNCTION FILE ENTRIES

This is exactly what you are allowed to do when you edit the FUNCTION file (#.5) using the Enter or Edit File Entries option.

To continue the above example, you could create a DASHDATE function which could then be used by any user to display date-valued fields and expressions in the DAY-MONTH-YEAR format as follows:

```
    Select OPTION: ENTER AND EDIT FILES
    INPUT TO WHAT FILE: FUNCTION
    EDIT WHICH ATTRIBUTE: ALL// <RET>
    Select COMPUTED-FIELD FUNCTION: DASHDATE
      ARE YOU ADDING 'DASHDATE' AS A NEW COMPUTED-FIELD FUNCTION? Y <RET>
(YES)
    MUMPS CODE: S X=+$E(X,6,7)_"-"_+$E(X,4,5)_"-"_$E(X,2,3)
    EXPLANATION: PRINTS DATE IN "DD-MM-YY" FORMAT
    DATE-VALUED: NO
    NUMBER OF ARGUMENTS: 1
    WORD-PROCESSING: <RET>
```

Notice that the MUMPS CODE field contains code to transform the variable X (the argument of the function) into a different X. If two arguments were required for the function, the first would be found in the variable X1 and the second in X. Although the new function being created here takes a date-valued argument, it is not itself considered to be date-valued since it doesn't produce values that look like the standard FileMan internal representation of a date. If this function was only

meaningful in a word processing context, you would put a W at the "WORD-PROCESSING:" prompt.

NOTE: If there is an output transform on a field, the function code is applied to the field after it has been transformed. In most cases, if a field has an output transform, you should therefore use the syntax FUNCTION_NAME(INTERNAL(FIELD_NAME)), rather than FUNCTION_NAME(FIELD_NAME).

A function can also be defined as taking no arguments. This is very similar to the special variables in M like $I and $H. For example, you could define a function like BELL as follows:

```
    Select COMPUTED-FIELD FUNCTION: BELL
       ARE YOU ADDING A NEW COMPUTED-FIELD FUNCTION? Y <RET>  (YES)
    MUMPS CODE: SET X=$C(7) <RET>    EXPLANATION: CAUSES A 'BEEP' TO OCCUR
ON OUTPUT <RET>   DATE-VALUED: NO
    NUMBER OF ARGUMENTS: 0
    WORD-PROCESSING: <RET>

Users could then embed "beeps" in output templates by entering:

    FIRST PRINT FIELD: BELL
```

NOTE: No parentheses are shown for a function with no arguments.

You can delete a function in the usual way by deleting the NAME of the function. Such deletions do not harm any computed fields that already have been created using the function. However, you may not edit the computed field unless you remove reference to the deleted function.

WARNING: Due to concatenation, do not use IF, FOR or QUIT statements when defining functions. Also, any variables you introduce within a function's code (but not X, X1, etc.) should be NEWed.

The Function file already contains several functions. Consult the "VA FileMan Functions" chapter of the *VA FileMan Advanced User Manual* for a description of the functions exported with VA FileMan.